

SPCM Lab-10

Objective : Creating a AWS RDS Instance in Terraform.

1. Create Terraform directory.

```
→ ~ mkdir terraform-rds
→ ~ cd terraform-rds
→ terraform-rds
```

2. Create terraform configuration file (main.tf) :

```
main.tf 2 x
main.tf > resource "aws_db_instance" "My-RDS"
1  provider "aws" {
2      region = "us-east-1"
3      access_key = "AKIA232UVZYDK5TANG62"
4      secret_key = "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"
5  }
6
7
8  resource "aws_db_instance" "My-RDS" {
9      allocated_storage      = 10
10     identifier              = "vidhantdb" //name of database
11     engine                  = "mysql"
12     engine_version          = "5.7"
13     instance_class          = "db.t3.micro"
14     username                = "admin"
15     password                = "admin123"
16     parameter_group_name    = "default.mysql5.7"
17     skip_final_snapshot     = true
18     publicly_accessible     = true //opens public access
19     tags = {
20         Name = "Myrdsdb"
21     }
22 }
```

3. Initialize, validate and Apply :

terraform init :

```
• → Lab-9 terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.37.0...
- Installed hashicorp/aws v5.37.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
```

terraform validate :

```
• → Lab-9 terraform validate
Success! The configuration is valid.

○ → Lab-9
```

terraform apply :

```
• → Lab-10 terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_db_instance.My-RDS will be created
+ resource "aws_db_instance" "My-RDS" {
+ address = (known after apply)
+ allocated_storage = 10
```

4. Verify Users in AWS console :

Databases (1)							
<input type="text" value="Filter by databases"/>							
<input type="checkbox"/>	DB identifier ▲	Status ▼	Role ▼	Engine ▼	Region & AZ ▼	Size ▼	Recomm
<input type="radio"/>	vidhantdb	Available	Instance	MySQL Community	us-east-1c	db.t3.micro	

Add inbound rules in security group to access RDS from local MYSQL workbench :

sg-0a0220e58a8f863f4

All traffic ▼

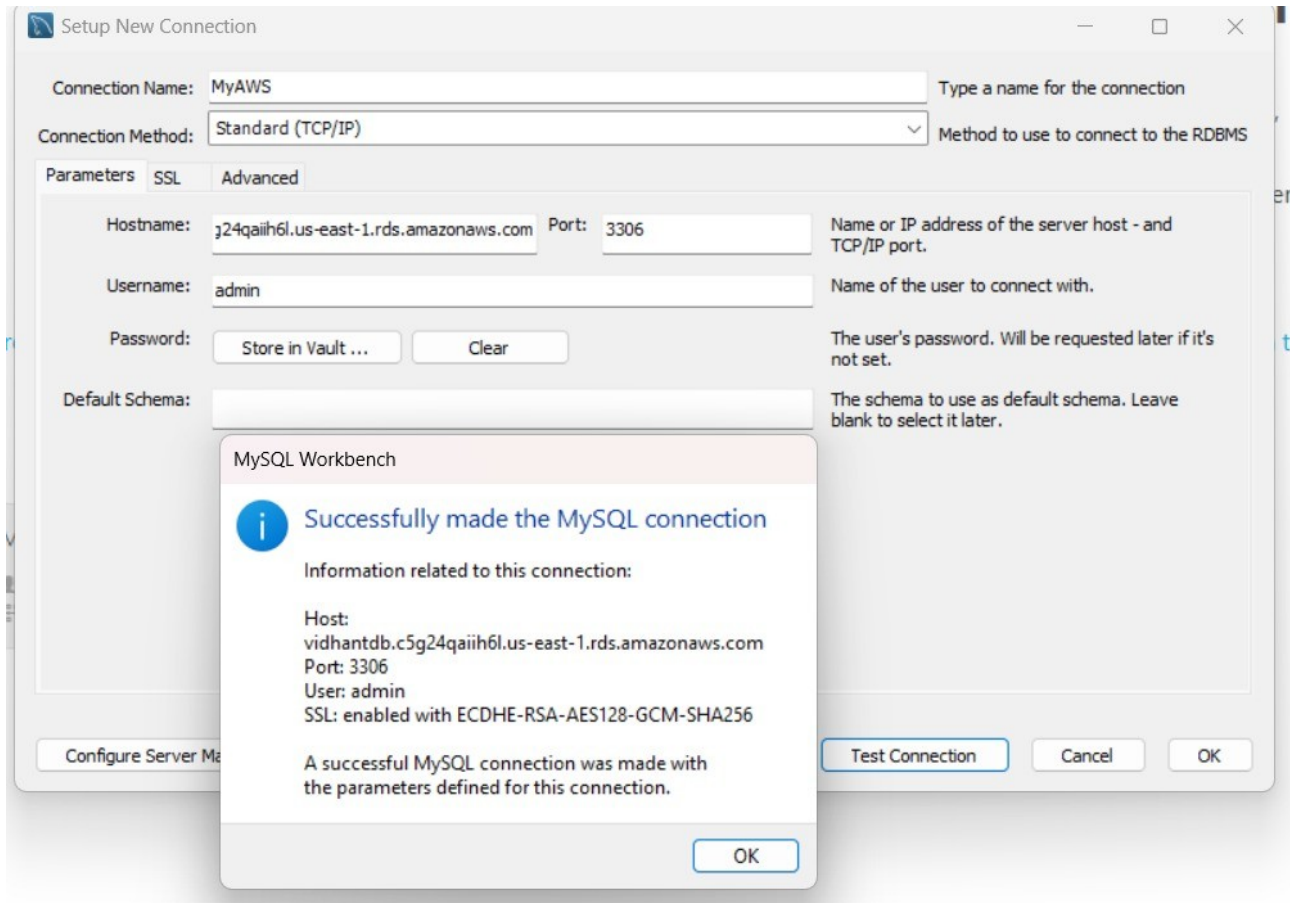
All

All

Custom ▼

Add rule

5. Connect RDS instance to MYSQL Workbench:



6. Clean up Resources (terraform destroy) :

```
o → Lab-10 terraform destroy -auto-approve
aws_db_instance.My-RDS: Refreshing state... [id=db-DVR0GM3GG3HZ7HX3LYCZDIPVM4]

Terraform used the selected providers to generate the following execution plan. Resource actions
indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

# aws_db_instance.My-RDS will be destroyed
- resource "aws_db_instance" "My-RDS" {
  - address = "vidhantdb.c5g24qaiih6l.us-east-1.rds.amazonaws.com"
```